Design and Implementation of VOD Database System

Jaegeol Yim, Gyeyoung Lee and Thanh C. Le

Dept. of Computer Engineering Dongguk university at Gyeongju
{yim, lky, lct}dongguk.ac.kr

Abstract

A database is an organized collection of data and the database data collection with DBMS is called a database system. A numerous database systems have been developed. In our previous work, we have introduced our design of database system for a VOD system. Nowadays computing power of mobile terminals is utterly astonishing and playing video on them became popular. So, this paper extends the previous work to mobile VOD and further discusses implementation of the design[1].

Keywords: *VOD, mobile, database, DBMS*

1. Introduction

Video on Demand (VOD) allows users to select or search for a video and watch it. VOD has been so widely used that the majority of Internet traffic is video data. In [1], the authors described an interactive VOD service, which supports VCR-like functions. It consists of two parts. The first is a database, which contains information related to the videos. The second part is the actual video distribution using the client-server model. The user can issue a query to the database and retrieve a collection of matching records. The user can then choose the link to the preferred video and watch it through the client application. To manage the database system the latter authors developed a client application that can be used to create or drop a table and insert, modify or delete video records and their associated information.

Web services are small and discrete building-block applications that can be used for building huge useful applications. By using web services, we can efficiently develop web applications. There are so many web service providers. Sabre[2], Datalex[3], and Galileo[4] provide web services for travel and tourism businesses. Amazon web service provides web services for commerce [5]. In [6], the authors introduced a Web-based Geo-spatial Service Platform using GML (Geography Markup Language) and Microsoft .NET.

The next generation database management systems (DBMS) should be a federation of distributed, heterogeneous and autonomous components. Such components constitute web database services. In [7], the authors presented the full spectrum of possible DBMSs based on the Service-Oriented Database Architecture (SODA). Then, the latter authors proposed one possible instance of SODA, called DBNet. The key ideas behind DBNet are as follows: (1) Publish query processing services (2) Discover the query processing services via UDDI and (3) Consume the query processing services and use them as part of a distributed query execution plan.

[1] Earlier version of this paper was presented in The 2012 International Conference on Information Science and Technology

Despite the fact that the condition of mobile computing has been largely improved in recent years, applying traditional Web services (i.e., SOAP-based services) models to mobile computing may result in unacceptable performance overheads [8]. SOAP-based services are heavy-weighted services that are not applicable to mobile services in comparison with the light weight RESTful services. Therefore, we develop RESTful web services for VOD DB systems.

2. Related Works

This paper is an extension of our paper presented in "The 2012 International Conference on Information Science and Technology." The readers are referred for the related works to it. [9] presents different types of presentation plans. A VOD system for elementary schools has been introduced in [10]. [11] proposes a new schema for achieving an online VOD system. [12] and [13] propose a distributed storage VOD system for delivering generated video content stored in IP-based user storage. [14] proposes a unique VOD architecture and implementation to achieve efficient QoS. A new proactive data replication mechanism is proposed and implemented in [15]. [16] developed CurioView which uses metadata and retrieval technology to satisfy the viewers' curiosity by recommending a wide-ranging video content related to the content the viewer is currently watching.

Several technologies, such as P2P [17], CDN [18] and CSMS [19] have been proposed and developed to enhance the performance of VoD systems. In [20], the authors present the design and implementation of a performance monitoring tool for clustered streaming media server systems. [6] and [21] proposed a geospatial data and services sharing framework based on Web portal technologies, Web services, OGC and W3C Standards. [7] presents the full spectrum of possible DBMSs based on such a Service-Oriented Database Architecture (SODA), and proposes one possible instance of SODA.

3. User Requirements

Many studies on developing VOD have been published [6, 10, 11, 22-24]. Considering the user's requirements of them, we propose comprehensive user requirements for VOD database systems. The detail of them will be discussed in Section 4.

4. Implementation of RESTful Services

Author names and affiliations are to be centered beneath the title and printed in Times New Roman 12-point, non-boldface type. Multiple authors may be shown in a two or three-column format, with their affiliations below their respective names. Affiliations are centered below each author name, italicized, not bold. Include e-mail addresses if possible. Follow the author information by two blank lines before main text.

We describe our implementation of web services in this section. A VOD database system usually consists of many tables. The tables included in our system are clearly shown in Figure 1. Web services for mobile programming should be based on RESTful, making them light weight and adequate to thin clients. Among the many ways of implementing RESTful web services, servlet of Java and WCF of .Net are common choices in practice. We are employing servlets to implement our RESTful web services. In RESTful web services, doGet( ) is the method to be executed when the caller invokes

the GET method to get some information. Similarly, doPost( )/deDelete( ) is the method to be executed when the caller invokes the POST/DELETE method to put/remove some information in/from the object.

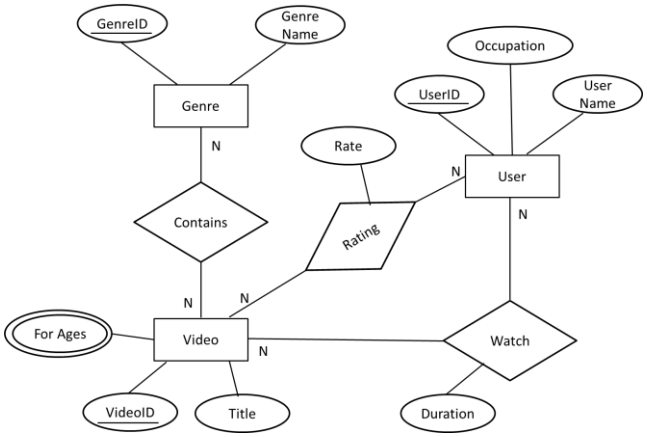

Figure 1. An E-R Diagram of our Database System

4.1 Directory Service

The main objective of 'directory service' is returning a list of all the genre names upon user. If the caller invokes the GET method of Genre Servlet, then doGET( ) of Genre is executed. It connects to the database and executes a query to retrieve all genre names by using the methods defined in DataLayer servlet; it then returns the list of genre names to the client.

4.2 Search Services

The main objective of 'search service' is returning a list of all the videos whose title, cast, description, or others match to the search phrase. If the caller invokes the GET method of search Servlet with a phrase such as 'lotus', then doGET( ) of Search is executed. It connects to the database and executes a query to retrieve all videos whose titles match the key phrase by using the methods defined in DataLayer servlet; it then returns the list of videos to the client.

4.3 Recommendation Service

A VOD system usually provides a recommendation service. There are many strategies of making recommendations: most popular videos, most recent videos, and so on. Among the generally used strategies, the user's personal preference-based recommendation is the most attractive. Therefore, we need the services of managing users' personal information (Users Service), recording users' history of watching videos (VideoUser Service), and recording users' evaluation of videos (Rating Service).

4.3.1 Users Service

This service helps users to add more users or update user's information. This service is also implemented as a servlet. This servlet retrieves the user's email address to check

whether or not the user exists in the database. If the user does not exist, then the servlet adds the user's information to the database, else it updates the user's information.

4.3.2 VideoUser Service

This service provides the function of adding a new record consisting of VideoID, UserID, TimeStart and TimeEnd. A record represents that the user (whose ID is UserID) watched the video (whose ID is VideoID) from TimeStart to TimeEnd. This service takes the arguments from the request and adds a record to VideoUser table

4.3.3 Rating Service

Together with the view history collected by 'VideoUser Service', users' evaluation is one of the most important resources for making recommendations. The 'Rating Service' provides two kinds of services, adding user's evaluation and retrieving the rate assigned by the user. Given VideoID, UserID and Rate, the former adds a new record or makes an update to the database. So, this service is implemented as a POST method. On the other hand, given VideoID and UserID, the latter returns the rate the user represented by UserID given to the video represented by VideoID. So, this service is implemented as a GET method.

4.3.4 Recommendation Service

Implementing a good recommendation system is very difficult, and it is out of the scope of this paper. However, a recommendation service should be included in our RESTful web services. So, we implement a simple one that returns a list of video titles that have been mostly watched by the subscribers who are similar to the current user. A subscriber is considered to be similar to the user if age, hobby, occupation, gender, and so on are the same. The current user's UserID is passed to this servlet; it finds all subscribers similar to the current user and calls them group A. Then, it returns a list of video titles that have been most frequently watched by group A.

4.4 DatabaseManagement Service

The purpose of this paper is to provide web services for VOD database systems. Therefore, the functions of creating a new database and table, deleting a database and table, and modifying a database and table should be provided by our services. Therefore, we implement a serlet named DatabaseManagement where the CreateDatabase, CreateTable, DeleteDatabase, DeleteTable methods are defined.

4.5 FileUpload Service

We cannot build up a VOD database system without video files. Therefore, we implement 'FileUpload' servlet, which can be used to upload a video file. The servlet takes PATH and the content of the video file to be uploaded as its parameter. Then, it saves the content at PATH.

5. Conclusion

Extending our previous work, we have introduced our design and implementation of database system for mobile VOD service. Our implementation is effective for mobile applications because it is RESTful.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0006942) and by 'Industry, Academy and Research Institute Co-development of Techniques' funded by the Small and Medium Business Administration (R00046281).

References

[1] I. Diamantis, A. Ntalakas, M. Strintzis, "Real time video distribution using publication through a database," International Symposium on Computer Graphics, Image Processing, and Vision, pp. 208 – 215 **(1998)**.

[2] Sabre, http://www.sabre.com.

[3] Datalex, http://www.datalex.com.

[4] http://xml.coverpages.org/GalileoGlobalWS.html.

[5] Amazon Web Services, http://aws.amazon.com/.

[6] Z. Guo, X. Wang and G. Sun, "Research and application on spatial data Web service based on .Net platform", IEEE International Geoscience and Remote Sensing Symposium, vol. 6, pp. 3709 – 3711 **(2003)**.

[7] W. Tok and S. Bressan, "DBNet: A Service-Oriented Database Architecture," 17th International Workshop on Database and Expert Systems Applications, pp. 727 – 731 **(2006)**.

[8] B. Upadhyaya, Y. Zou, H. Xiao, J. Ng and A. Lau, "Migration of SOAP-based services to RESTful services " 13th IEEE International Symposium on Web Systems Evolution, pp. 105 – 114 **(2011)**.

[9] E. Hwang, B. Prabhakaran and V. Subrahmanian, "Presentation planning for distributed VoD systems," IEEE Transactions on Knowledge and Data Engineering, 14, 5, pp. 1059 – 1077 **(2002)**.

[10] C. Li and H. Cui, "Design and implementation of video-on-demand system for elementary and secondary schools," IEEE International Conferences on Multimedia Technology, pp. 4891 – 4893 **(2011)**.

[11] F. Jin "Design and Implementation Video on Demand System Based on FMS +FLV," Fourth International Conference on Genetic and Evolutionary Computing, pp. 398 – 401 **(2010)**.

[12] Y. Nomoto, T. Kuzukawa and Y. Ishibashi, "Distributed storage VOD system using XDMS: Implementation and performance evaluation," IEEE Region 10 Conference, pp.1438–1443 **(2010)**.

[13] Y. Nomoto, T. Kuzukawa and Y. Ishibashi, "IP storage and stored content management using SIP presence server with XML database," Australasian Telecommunication Networks and Applications Conference, pp. 1–6 **(2011)**.

[14] M. Dakshayini, H. Guruprasad, H. Maheshappa and A. Manjunath, "Load Balancing in Distributed VoD Using Local Proxy Server Group [LPSG]," International Conference on Computational Intelligence and Multimedia Applications, Vol. 4, pp.162–168 **(2007)**.

[15] X. Liao, F. Zhang, H. Jin and L. Yu, "iDARE: Proactive Data Replication Mechanism for P2P VoD System," IEEE 10th International Conference on Computer and Information Technology, pp 682 – 689 **(2010)**.

[16] H. Sumiyoshi, M. Sano, J. Goto, T. Mochizuki, M. Miyazaki, M. Fujii, M. Shibata and N. Yagi, "CurioView: TV recommendations related to content being viewed," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, pp. 1 – 6 **(2010)**.

[17] E. Lua, J. Crowcroft, M. Pias, et al., "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys & Tutorials, vol. 7, pp. 72-93 **(2005)**.

[18] J. Broberg, R. Buyya and Z. Tari, "MetaCDN: Harnessing 'Storage Clouds' for high performance content delivery," Journal of Network and Computer Applications, vol. 32, pp. 1012-1022 **(2009)**.

[19] Y. Chai, Z. Du and Y. Chen, "A stepwise optimization algorithm of clustered streaming media servers," Journal of Systems and Software, vol. 82, pp. 1344-1361 **(2009)**.

[20] X. Wang, X. Liu, H. Xie, Z. Du and L. Jin, "Design and Implementation of a Performance Monitoring Tool for Clustered Streaming Media Server Systems," International Symposium on Information Science and Engineering (ISISE), pp. 314 – 318 **(2010)**.

[21] Y. Jie, P. Yue and J. Gong, "Integration of geospatial Web services and Web portal technologies for geospatial Information sharing and processing," 17th International Conference on Geoinformatics, pp. 1 – 4 **(2009)**.

[22] S. Shim and Y. Lee, "Interactive TV: VoD meets the Internet," IEEE Journals Computer, Vol. 35, Issue 7, pp. 108 – 109 **(2002)**.
[23] S. Hameed, A. Alam, N. Nuh, N. Salim, "Integrated medical emergency model: An interactive web-based database," International Conference on Computer and Communication Engineering, pp.1–6 **(2010)**.
[24] http://www.grouplens.org/node/73.

Authors

Jaegeol Yim

Jaegeol Yim received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Engineering at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.

Gyeyoung Lee

Gyegoung Lee received the M.S. degree in Computer Science from the Dongguk University in 1982 and the Ph.D. degree in Computer Engineering from the Dankook University in Korea in 1992, respectively. He is a Professor in the Department of Computer Engineering at Dongguk University at Gyeongju, Korea. His current research interests include Petri net theory, AI systems and Speech processing systems.

Thanh C. Le

Thanh Le is a graduate student at Computer Science Department of Dongguk University from 2011. His current research interests include Artificial Intelligent, Natural Language Processing, Dialog Systems, Statistics, and Probability.